

Though the discussion above is directed to assignment of interconnect fabric ports to virtual SANs, those skilled in the art will appreciate that the techniques are equally applicable to assignment of storage devices or other SAN components seen by the hosts.

## 5 *Maintaining and Updating SAN History Data*

As noted above, in the illustrated embodiment, the SAN manager stores an internal model store 125 of the SAN topology. As illustrated in FIGURE 25, that model store contains objects 126 representing components of the SAN (e.g., hosts 12, storage devices 14, interconnect element 16a), their attributes and the interrelationships therebetween (e.g., assignment and/or accessibility of a host 12 to a storage device 14). These objects can be arranged hierarchically or otherwise (e.g., via link lists or other associations) to reflect relationships among the SAN components.

In the illustrated embodiment, the objects are object-oriented programming "objects," though other programming constructs can be used in addition or instead. Moreover, in the illustrated embodiment, the objects are maintained both in a persistent database, as well as in a runtime form (e.g., in the random access memory of manager 20).

The SAN manager 20 additionally includes a historical model store 128 that reflects a one-deep history (or, in alternate embodiments, still deeper) about *specific* components and/or relationships within the SAN – to wit, components and/or relationships that have recently changed. This information is used to during generation of displays enumerating (e.g., listing) the

SAN componentry and/or showing its topology (collectively, "topology"), e.g., on the administrator console 52.

Specifically, in the illustrated embodiment, it is used to identify (by way of non-limiting example, via highlighting, graying out or otherwise altering the appearance of) graphical objects representing components and/or relationships that are new, missing, broken, need attention, have a changed attribute, or have attained a "suspect" status, e.g., since the time of the last generated display -- and, more precisely, since the time the operator/administrator last asked that such highlighting, graying-out or other identifications be cleared.

Such a display is depicted in FIGURE 26. Shown there is a hierarchical display 151 of the type presented on the operator/administrator console. This includes a graphical object (e.g., icons) representing the SAN as a whole, to wit, element 153, and graphical objects representing the components thereof, here illustrated as Components 1 – 6 (elements 155, 157, 159, 161, 163, 165). It will be appreciated that the specific form of the display can be varied depending on operator preferences and needs. Moreover, it will be appreciated that representations other than graphical objects (e.g., text labels, and so forth) may be used.

In the illustration, Component 3 (element 159) and Component 6 (element 165) are color-coded to indicate that they were newly added to the SAN since the last console presentation to the operator/administrator and/or since the he last cleared the updates. Component 4 (element 161) is identified as missing (e.g., and likely removed from the SAN), while Component 2 (element 157) is identified as suspect. In the illustrated embodiment, a component is deemed "suspect" if

its status has been reported inconsistently among the scans in which it appears. Though color coding (or shading) is used in the illustrated embodiment, it will be appreciated that any range of visual, aural or other sensory indicators can be employed to identify the status of displayed, updated components (e.g., Components 2, 3, 4, 6).

5

In contrast to having every object in model store 125 maintain status history for its respective component, reference objects 130 (hereinafter, "HistoryData" objects) are (instantiated and) maintained in the store 128 for only those SAN components whose statuses have changed, e.g. since the time last displayed to the operator/administrator and/or since the he last cleared the updates. In the illustrated embodiment, each HistoryData object 120 includes a unique identifier referencing the SAN object 126 to which it pertains, and further includes an indicator of the status of the underlying component (e.g., "new", "missing", "broken", "moved", "needs attention", "attribute change" or "suspect"). Those skilled in the art will appreciate that other embodiments may use other statuses in addition or instead (e.g., modified, offline, format degraded, etc.) It will also be appreciated that the HistoryData object may maintain additional information (e.g., time stamps, etc.) Moreover, it will be appreciated that in the illustrated embodiment, no HistoryData object is maintained for objects (and underlying components) in model 125 whose status is "Normal".

20 As above, the HistoryData objects can be object-oriented programming "objects" or other constructs suitable for these purposes. Also as above, the HistoryData objects are preferably stored in a persistent manner, as well in a runtime form.